Opprette bruker:

```
If(
    IsBlank(CUA_NameI.Text) ||
    IsBlank(CUA_Mail.Text) ||
    IsBlank(CUA_LastName.Text) ||
    IsBlank(CUA_Password.Text) ||
    IsBlank(CUA_ConfirmPassword.Text) ||
    IsBlank(CUA_Firm.Text) ||
    IsBlank(CUA_FirmMail.Text) ||
    IsBlank(CUA_PhoneNumber.Text),

    //Sjekker at ingen av feltene er blanke

    Notify(
        "Vennligst fyll ut alle feltene før du oppretter en bruker.",
        NotificationType.Error
    ),
    If(
        CUA_Password.Text = CUA_ConfirmPassword.Text,       //sjekker at passordene er like
        Patch(      //bruker patch til å opprette en ny bruker
            'New Users',
            Defaults('New Users'),      //for å si at en ny rad skal opprettes
            {
                Name: CUA_NameI.Text,
                Username: CUA_Mail.Text,
                Password: CUA_Password.Text,
                'Confirm Password': CUA_ConfirmPassword.Text,
                Firm: CUA_Firm.Text,
                'Firm Email': CUA_FirmMail.Text,
                'Phone Number': CUA_PhoneNumber.Text
            }
        );
        Notify(
            "Bruker opprettet.",
            NotificationType.Success
        );
        Set(        //samme som i vanlig login setter en global bruker
            gblCurrentUser,
            LookUp(
                'New Users',
                Username = CUA_Mail.Text && Password = CUA_Password.Text
            )
```

Ny forespørsel:

```
If(
    IsBlank(NRC_RN.Value) ||        //sjekker at alt er fylt ut
    IsBlank(NRC_CV.Value) ||
    IsBlank(NRC_CT.Selected.Value) ||
    IsBlank(NRC_SD.SelectedDate) ||
    IsBlank(NRC_ED.SelectedDate) ||
    IsBlank(NRC_CR.Selected.Value) ||
    IsBlank(NRC_CQ.Value) ||
    IsBlank(NRC_IC.Selected.Value),

    Notify(
        "Please fill out all the information.",
        NotificationType.Error
    ),

    Patch(
        'Charter Requests',
        Defaults('Charter Requests'),       //vi skal opprette ny rad
        {
            Name: Text(NRC_RN.Value),
            'Contract Value': Value(NRC_CV.Value),
            'Cargo Quantity': Value(NRC_CQ.Value),
            'Start Date': NRC_SD.SelectedDate,
            'End Date': NRC_ED.SelectedDate,
            'Logged in user': gblCurrentUser,       //henter ut hvem som sender forespørsel rett fra logget inn kunde
            Firm: gblCurrentUser,
            'Charter Type': NRC_CT.Selected.Value,
            'IMO Classification': NRC_IC.Selected.Value,
            'Canal requirement': Table(NRC_CR.SelectedItems)
        }
    );
```

Sikre robusthet:

```
Switch(
    Text(NRC_IC.Selected.Value),
    "IMO I", 14800,
    "IMO II", 47000,
    "IMO III", 47200,
    0
)
```

```
Switch(
    Text(NRC_IC.Selected.Value),

    "IMO I",
    If(
        Value(NRC_CQ.Value) > 14800, // Sjekker om verdien er for høy
        Notify("Our IMO I fleet cant facilitate cargo quantities larger than 14 800 tons", NotificationType.Warning)
    ),

    "IMO II",
    If(
        Value(NRC_CQ.Value) > 47200,
        Notify("Our IMO II fleet cant facilitate cargo quantities larger than 47 200 tons", NotificationType.Warning)
    ),

    "IMO III",
    If(
        Value(NRC_CQ.Value) > 47200,
        Notify("Our IMO III fleet cant facilitate cargo quantities larger than 47 200 tons", NotificationType.Warning)
    )
)
```

Filtrere forespørsler:

```
Switch(
    RROB_CustomerRequestDropdown.Selected.Value,
    // Hvis "All" er valgt, vis hele listen
    "All",
    'Charter Requests',
    // Hvis "Unanswered" (Ikke avslått OG ingen båt tildelt)
    "Unanswered",
    Filter(
        'Charter Requests',
        Declined=false && IsBlank('Allocated Vessel')
    ),
    // Hvis "Accepted" (Båt er tildelt)
    "Accepted",
    Filter(
        'Charter Requests',
        !IsBlank('Allocated Vessel')
    ),
    // Hvis "Declined" (Avslått er sant)
    "Declined",
    Filter(
        'Charter Requests',
        Declined = true
    ),
    // Standardvalg hvis ingenting matcher (valgfritt, her viser den alt)
    'Charter Requests'
)
```

Filtrere skip til forespørsler

```
If(
    RROB_CustomerRequestGallery.Selected.Declined = true,
    Blank(), //om request er declined er den blank
    With(
        {
            req: RROB_CustomerRequestGallery.Selected,      //setter noen variabler for oversiktelighet
            allocatedVessel: RROB_CustomerRequestGallery.Selected.'Allocated Vessel',
            cont: Contracts
        },
        If(
            !IsBlank(allocatedVessel), //om den valgte forespørselen har et skip

                    //vi filtrerer Vessel-tabellen for å finne skipet som matcher det tildelte skipet
            Filter(
                'Vessel',
                Name = RROB_CustomerRequestGallery.Selected.'Allocated Vessel'.Name
            ),

            With(        //kjører når den valgte forespørselen IKKE har fått skip eller er declined
                {
                    imoReq: Text(req.'IMO Classification'),

                    relevantCanalSpecs: If(
                        IsEmpty(req.'Canal requirement'),
                        Blank(),
                        Filter(
                            'Canal Requirements',
                            Text('Canal Requirement Code') in ForAll(       //forsøker å hente ut kanalkravene
                                req.'Canal requirement',
                                Text(Value)
                            )
                        )
                    )
                },
```

```
//forsøker å beregner de strengeste kravene
With(
    {
        limitAirDraft: Coalesce(Min(relevantCanalSpecs,'Max Air Draft'), 99999),
        limitBeam: Coalesce(Min(relevantCanalSpecs,'Max Beam'), 99999),
        limitDraft: Coalesce(Min(relevantCanalSpecs,'Max Draft'), 99999),
        limitLOA: Coalesce(Min(relevantCanalSpecs,'Max LOA'), 99999),
        reqMarpol: !IsBlank(LookUp(relevantCanalSpecs,'Requires MARPOL' = true)),
        reqSolas: !IsBlank(LookUp(relevantCanalSpecs,'Requires SOLAS' = true))
    },
    Filter(
        'Vessel',
        // 1) Kapasitet
        IsBlank(req.'Cargo Quantity') || (DWT * 0.8) >= req.'Cargo Quantity',
        // 2) IMO-krav
        IsBlank(imoReq) || Switch(
            imoReq,
            "IMO I", Text('IMO Classification') = "IMO I",
            "IMO II", Text('IMO Classification') = "IMO I" || Text('IMO Classification') = "IMO II",
            "IMO III", Text('IMO Classification') = "IMO I" || Text('IMO Classification') = "IMO II" || Text('IMO Classification') = "IMO III",
            true
        ),
        // 3) Kanal-krav
        IsBlank(relevantCanalSpecs) || (
            'Air Draft (Høyde)' <= limitAirDraft &&
            'Beam (Width)' <= limitBeam &&
            'Draft (Dypgang)' <= limitDraft &&
            'Length Overall (LOA)' <= limitLOA &&
            (!reqMarpol || 'MARPOL sertifisert' = true) &&
            (!reqSolas || 'SOLAS sertifisert' = true)
        ),
        // 4) Ingen nåværende kontrakt for skipet
        IsBlank(
            LookUp(
                cont,
                'Vessel ID' = Text(ThisRecord.Vessel_contract) && StartDate_Contract <= Today() && EndDate_Contract >= Today()
            )
        )
    )
```

Tildele skip:

```
With(
    {
        wReq: RROB_CustomerRequestGallery.Selected,
        wVessel: RROB_PossibleShipsAllocated.Selected
    },
    If(
        IsBlank(wReq),
        Notify("Please select a request first.", NotificationType.Error),

        If(
            IsBlank(wVessel),
            Notify("Please select a vessel to allocate.", NotificationType.Error),

            //dersom alt er fylt ut kjøres det
            Concurrent(     //ber den opprette kontrakt og oppdatere forespørsel samtidig
                //oppretter kontrakten
                Patch(
                    Contracts,
                    Defaults(Contracts),
                    {
                        Name_Contract: "Contract: " & wReq.Name,
                        StartDate_Contract: wReq.'Start Date',
                        EndDate_Contract: wReq.'End Date',
                        Vessel_contract: Text(wVessel.'Vessel ID'),
                        Customer_contract: Text(wReq.'Logged in user'.'User-ID')
                    }
                ),

                //oppdaterer den valgte forespørselen
                Patch(
                    'Charter Requests',
                    wReq,
                    {
                        'Allocated Vessel': wVessel,
                        Declined: false
                    }
                )
            );

            Notify("Contract created successfully!", NotificationType.Success);
```